Reamostrando imagens

- Link do post https://www.instagram.com/p/CkyjEIypYae/ :

```
# Bibliotecas
import numpy as np
import tifffile as tif
import matplotlib.pyplot as plt
from skimage.transform import rescale
from spectral import imshow

# Função para redimensionar imagens
def resample(source, target, scale):

  scaled = rescale(source, (scale,scale))

  if target.shape[0] < scaled.shape[0]:
    scaled = scaled[:target.shape[0],:]
  else:
    target = target[:scaled.shape[0],:]

  if target.shape[1] < scaled.shape[1]:
    scaled = scaled[:,:target.shape[1]]
  else:
    target = target[:,:scaled.shape[1]]

  return scaled

# Lendo arquivos
img = tif.imread('/content/drive/MyDrive/Dados_PDI/CUBO_GEOBIA.tif')
vv = tif.imread('/content/S1A_VV.tif')
vh = tif.imread('/content/S1A_VH.tif')

# Reamostrando
vv_20 = resample(vv, img[:,:,0], 0.6)
vh_20 = resample(vh, img[:,:,0], 0.6)
```

```
# Empilhando dados
stack = np.dstack([img, vv_20, vh_20])


print('Dimensões do cubo: ', img.shape)
print('Dimensões da cena VV: ', vv.shape)
print('Dimensões da cena VH: ', vh.shape)
print('Dimensões da cena VV reamostrada: ', vv_20.shape)
print('Dimensões da cena VH reamostrada: ', vh_20.shape)
```

Expansão do histograma

```
# Bibliotecas
import numpy as np
import tifffile as tif
import matplotlib.pyplot as plt
from spectral import imshow

#Função de expansão
def expansao(img, percent_ini, percent_fim):
    s = np.zeros_like(img)
    x,y = 0,255
    w = np.percentile(img, percent_ini)
    z = np.percentile(img, percent_fim)
    p = x + (img - w) * (y - x) / (z - w)
    p[p<x] = x
    p[p>y] = y
    s = p
    return s

#Leitura d imagem
b3 = tif.imread('B3.tif')

imshow(b3)# Visualização

#Cálculo da expansão e visualização
s = expansao(b3, 2, 98)
imshow(s)

b3_flat = b3.flatten()#Transformando imagem em vetor 1D
plt.hist(b3_flat, bins=200)
plt.show()

s_flat = s.flatten()
plt.hist(s_flat, bins=200)
plt.show()
```

Segmentando imagens

- Link do post https://www.instagram.com/p/ClD-GYLOh4I/

```
!pip install -q rasterio

# Bibliotecas
import numpy as np
from skimage import exposure
from skimage.segmentation import mark_boundaries, felzenszwalb, slic,
quickshift
import rasterio as rio
import matplotlib.pyplot as plt
import tifffile as tif

# Lendo imagem como array
src = rio.open('/content/drive/MyDrive/Curso PDI com
Python/cubo_s2.tif')

list_band = [src.read(i+1) for i in range(src.count)]

img = np.dstack(list_band)

# Armazenando metadados da imagem original
meta = src.profile

# Reamostrando para valoers entre -1 e 1
img2 = exposure.rescale_intensity(img)

# Segmentadores
quick = quickshift(img2[:,:,:3].astype('double'), kernel_size=5,
max_dist=1000,
                   ratio=0.1)
slic_ = slic(img2, n_segments=5000, compactness=0.1, sigma=1,
start_label=1)
felzen = felzenszwalb(img2, scale=1, sigma=1, min_size=70)

# Função de expansão
def expansao(img, percent_ini=2, percent_fim=98):
    s = np.zeros_like(img)
    x,y = 0,1
    w = np.percentile(img, percent_ini)
    z = np.percentile(img, percent_fim)
    p = x + (img - w) * (y - x) / (z - w)
    p[p<x] = x
    p[p>y] = y
    s = p
    return s

# Visualizando resultado
```

```
fig, ax = plt.subplots(2, 2, figsize=(15, 15), sharex=True,
sharey=True)

ax[0, 0].imshow(mark_boundaries(expansao(img2[:,:,7]), felzen))
ax[0, 0].set_title("Felzenszwalb", fontsize=15)
ax[0, 1].imshow(mark_boundaries(expansao(img2[:,:,7]), slic_))
ax[0, 1].set_title('SLIC', fontsize=15)
ax[1, 0].imshow(mark_boundaries(expansao(img2[:,:,7]), quick))
ax[1, 0].set_title('Quickshift', fontsize=15)
ax[1, 1].imshow(expansao(img2[:,:,7]), cmap="gray")
ax[1, 1].set_title('Original', fontsize=15)

for a in ax.ravel():
    a.set_axis_off()

plt.tight_layout()
plt.show()
```

Criando planos de informação

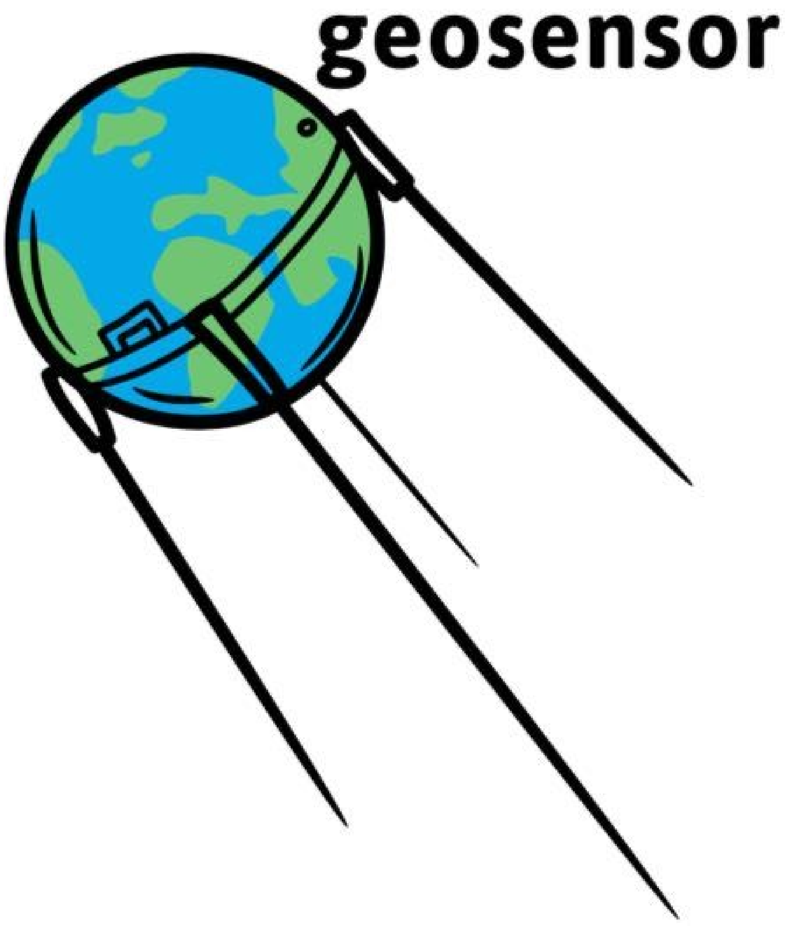
- Link do post https://www.instagram.com/p/CkfzfaeOM7U/

```
# Bibliotecas
import numpy as np
import tifffile as tif
import matplotlib.pyplot as plt
from skimage.transform import rescale
import cv2

# Lendo imagens
img = tif.imread('L71221071_07120010720_DN.tif')
sar = tif.imread('S1_DF_GRD.tif')

# Função para redimensionar imagens
def resample(source, target, scale):

  scaled = rescale(source, (scale,scale))

  if target.shape[0] < scaled.shape[0]:
    scaled = scaled[:target.shape[0],:]
  else:
    target = target[:scaled.shape[0],:]

  if target.shape[1] < scaled.shape[1]:
    scaled = scaled[:,:target.shape[1]]
  else:
    target = target[:,:scaled.shape[1]]

  return scaled
```

```
#Função de expansão
def expansao(img, percent_ini=2, percent_fim=98):
    s = np.zeros_like(img)
    x,y = 0,255
    w = np.percentile(img, percent_ini)
    z = np.percentile(img, percent_fim)
    p = x + (img - w) * (y - x) / (z - w)
    p[p<x] = x
    p[p>y] = y
    s = p
    return s

# Criando planos de informação
ndvi = (img[:,:,3] - img[:,:,2] )  /  ( img[:,:,3] + img[:,:,2])
sobelx = cv2.Sobel(src=ndvi, ddepth=-1,  dx=1, dy=0, ksize=5)
sobely = cv2.Sobel(src=ndvi, ddepth=-1, dx=0, dy=1, ksize=5)
vv = resample(sar[:,:,0], ndvi, 1/3)
vh = resample(sar[:,:,1], ndvi, 1/3)
razao = vv/vh

# Empilhando planos de informação
stack_features = np.dstack([ndvi, sobelx, sobely, vv, vh, razao])

lista_nomes= ['ndvi', 'sobelx', 'sobely', 'vv', 'vh', 'razao']

#Plotando em um gráfico de duas linhas e três colunas
fig,axes = plt.subplots(2,3,figsize=(15,10),sharex='all',
sharey='all')
fig.suptitle('Planos adicionais', fontsize=20)
axes = axes.ravel()
for i in range(stack_features.shape[2]):
  axes[i].imshow(expansao(stack_features[:,:,i]),cmap='gray')
  axes[i].set_title(lista_nomes[i],fontsize=15)
  axes[i].axis('off')
```

Outros tópicos importantes

- Erros comuns em Python https://www.instagram.com/p/CkIsFYlgWWU/
- Dica de leitura 1 https://www.instagram.com/p/CklaOPGulUc/
- Dica de leitura 2 https://www.instagram.com/p/Ck3qokSp-qV/
- Download de imagens Sentinel2 https://www.instagram.com/p/CjF1DWrOXqL/
- Geopandas 1 https://www.instagram.com/p/CUXo7IFLh89/
- Geopandas 2 https://www.instagram.com/p/CUpjBqyLrEL/
- Geopandas 3 https://www.instagram.com/p/CVNmdHwrJuk/
- Geopandas 4 https://www.instagram.com/p/CVgpDq7sJ-m/
- Geopandas 5 https://www.instagram.com/p/CVyEcVbP-8_/
- Github https://github.com/Gustavoohs


geosensor